

Memo

Datum [03-05-10](#)
Aan StUF Community
Van Maarten van den Broek
Onderwerp Ontwerpkeuzen bij het verStUffen van het RGBZ
Versie '1.043

Inhoudsopgave

1 Inleiding.....	2
2 Ontwerpbeslissingen.....	3
2.1 Het platslaan van relaties.....	3
2.2 Het definiëren van relaties en hun gerelateerden.....	5
2.3 Het expliciteren van semantiek.....	6
3 De StUF-entiteittypen en hun relatiegrafieken.....	8
3.1 Besluit en besluytype.....	8
3.2 Document (enkelvoudig en samengesteld) en zaakdocument.....	8
3.3 Medewerker en organisatorische eenheid.....	10
3.4 Zaak, zaaktype, status en statustype.....	11
4 Lijst met mnemonics.....	15

Versiehistorie

'1.04	Resultaten van bespreking in Expertgroep 21-4-2010 verwerkt.
'1.03	Een aantal discussiepunten over de modellering expliciet opgenomen in dit document als input voor de discussie in de expertgroep. Voor niet alle discussiepunten is al een keuze gemaakt.
'1.02	Tabelentiteittype voor Zaaktype Generiek (ZTG) toegevoegd en het omgaan met het tabelentiteittype voor Documenttype Generiek (DTG) aangepast. Verwijzing naar versie RGBZ gewijzigd in 0.9.4.
'1.01	Expliciet aangeven in de inleiding dat zkn0310 bindt aan protocolbindingen 0301

1 Inleiding

Dit document licht een aantal ontwerpbeslissingen toe, die genomen zijn bij het maken van het sectormodel zkn0310 voor het RGBZ versie 0.9.4. Het gaat hierbij bijvoorbeeld om het samenvoegen van meerdere RGBZ objecttypen in één StUF entiteittype, om het al dan niet opnemen van relatiesoorten uit het RGBZ in een StUF-entiteittype of om het niet opnemen van een relatiesoort in het RGBZ als een relatie-entiteittype binnen een StUF-entiteittype, maar als een 'platgeslagen' verzameling elementen in het StUF-entiteittype.

Dit document met ontwerpkeuzen is niet normatief. Het geeft een toelichting en motivatie bij een groot aantal gemaakte keuzen. De formele specificatie van het sectormodel zkn0310 ligt vast in de schema's zkn0310.ent.xsd, zknbg0310.ent.xsd, zknstuf0310.xsd en zkn0310.msg.xsd en de wsdl's zkn0310.beantwoordVraag.wsdl, zkn0310.ontvangAsynchroon.wsdl en zkn0310.verwerkSynchroneKennisgeving.wsdl. De methodiek voor het maken van schema's zal nog worden vastgelegd in een best practise voor het maken van schema's voor StUF sectormodellen. Op dit moment is dit best practise document nog niet beschikbaar. Het zkn0310.ent.xsd schema bevat de complexTypes voor de StUF-entiteitstypen gebruikt in de berichten of in andere StUF-entiteitstypen. Het schema zkn0310.msg.xsd bevat de definities van de elementen voor de verschillende berichten in zkn0310. Het schema zknstuf0310.xsd bevat restrictions op in het schema voor de StUF-standaard zelf gedefinieerde typen zoals de stuurgegevens en parametersVraag complexTypes. Het schema zknbg0310.ent.xsd bevat de restrictions op de basistypen uit het sectormodel bg0310 die ook binnen zkn0310 worden gebruikt.

Het sectormodel zkn0310 maakt gebruik van de protocolbindingen versie 0301. De servicedefinities voor de standaard SOAP/WUS protocolbinding zijn vastgelegd in de wsdl's zkn0310.beantwoordVraag.wsdl, zkn0310.ontvangAsynchroon.wsdl en zkn0310.verwerkSynchroneKennisgeving.wsdl. Voorschriften voor het maken van deze wsdl's staan in het document Protocolbindingen 0301.

De uitgangspunten voor het maken van het sectormodel zkn0310 waren hetzelfde als voor bg0310. Voor deze uitgangspunten wordt verwezen naar hoofdstuk 2 in het document keuzenVerStUffing RSGB dat onderdeel is van het sectormodel bg0310. Ook voor zkn0310 is gekozen voor binding aan https binnen de StUF-binding, omdat zaakberichten vaak privacygevoelige informatie bevatten.

In hoofdstuk 2 van dit document wordt in algemene zin ingegaan op enkele ontwerpissues bij het omzetten van een semantisch model naar hiërarchische berichtstructuren. Bij de bespreking van de verschillende StUF-entiteitstypen in hoofdstuk 3 worden de voor dat StUF-entiteittype gemaakte keuzen nader toegelicht. In hoofdstuk 3 van dit document wordt ingegaan op het omzetten van de objecttypen met een netwerkstructuur in het RGBZ naar StUF entiteitstypen en hun hiërarchie van relaties in StUF-berichten. Voor een toelichting op het gebruik van de relatiegrafieken wordt verwezen naar de inleiding van hoofdstuk 3 in het document keuzenVerStUffingRSGB. Het document wordt in hoofdstuk 4 afgesloten met omschrijvingen van de gebruikte mnemonics.

2 Ontwerpbeslissingen

2.1 Het platslaan van relaties

Het semantisch model definieert als fundamenteel objecttype elk relevant begrip in het domein met voorkomens. Vaak zijn de voorkomens van een objecttype onafhankelijk van andere voorkomens, maar niet altijd. Een orderregel komt bijvoorbeeld alleen maar voor als onderdeel van een order [en hetzelfde geldt voor de status van een zaak](#). In een semantisch model worden naast attribuutsoorten van een objecttype ook relatiesoorten gedefinieerd. Het RSGB definieert bijvoorbeeld als objecttype Land en definieert het land waar een persoon uit afkomstig is als een relatie van het objecttype Natuurlijk Persoon naar het objecttype Land. Gemeenten, wijken en buurten worden in het RSGB gemodelleerd als afzonderlijke objecttypen met relaties die aangeven dat een buurt in een wijk ligt en een wijk in een gemeente. Het RSGB legt vanuit Benoemd object een relatie naar Buurt om aan te geven in welke buurt een benoemd object ligt.

Een relatie is in StUF een ingewikkelder constructie dan een element voor een attribuutsoort binnen een objecttype. Een attribuutsoort wordt in StUF gerepresenteerd door een enkelvoudig (simple) element binnen een entiteittype. Een relatiesoort wordt gerepresenteerd door een samengesteld (complex) element. Het relatie-element bevat het element voor de gerelateerde en binnen de gerelateerde moeten weer minstens de kerngegevens voorkomen. De relatie en gerelateerde hebben minimaal als attribute entiteittype en in kennisgevingen ook verwerkingssoort. Wanneer historie relevant is, dan dient in de relatie ook nog het element tijdvakRelatie te worden opgenomen om te kunnen aangeven hoelang de relatie bestond. Het opnemen van de identificatie van een gerelateerde als enkelvoudig element in [eenhet](#) entiteittype [vanwaaruit de relatie ligt](#) is eenvoudiger. Het element loopt voor zijn historie dan gewoon mee met de overige elementen. In geval van het immigratieland is bijvoorbeeld het opnemen van de landcode binnen NPS veel eenvoudiger dan het opnemen van een relatie 'isGeïmmigreerdUit' naar LND. Het opnemen van elementen van een gerelateerde in het entiteittype vanwaaruit de relatie ligt wordt ook wel het platslaan van een relatie genoemd.

Het platslaan van relaties heeft neveneffecten en dient daarom met zorg te worden toegepast. De belangrijkste oorzaak van neveneffecten is het opnemen van 'beschrijvende' attribuutsoorten naast strikt identificerende attribuutsoorten. Dan rijst de vraag hoe die beschrijvende gegevens onderhouden moeten worden. Dit kan zowel vanuit kennisgevingberichten voor de gerelateerde van de platgeslagen relatie als vanuit berichten voor het entiteittype waarin de gegevens zijn platgeslagen.

Een sprekend voorbeeld is het opnemen van de naam van de woonplaats waarin een openbare ruimte is gelegen en van de naam van de openbare ruimte naast het opnemen van de identificatie van die openbare ruimte in een adres. In de praktijk is dit noodzakelijk, omdat lang niet elk systeem dat berichten genereert de openbare ruimte identificatie vanuit de BAG kent (denk bijvoorbeeld aan een webapplicatie waar de burger zelf adresgegevens invult).

Dit voorbeeld roept twee vragen op:

1. Als er historie van een adres wordt getoond [vanuit het entiteittype waarin het adres is platgeslagen](#), moet dan ook getoond worden dat de openbareRuimtenaam en de woonplaatsnaam in de loop van de tijd gewijzigd zijn of wordt er alleen historie getoond als het adres gekoppeld wordt aan een andere openbareRuimte of woonplaats (gebeurt in de praktijk veel minder vaak dan een vernaming).

2. Als de openbare ruimte of de woonplaats een andere naam krijgt, wordt dit dan gecommuniceerd door middel van een kennisgeving voor adres of door middel van kennisgevingen voor OpenbareRuimte en Woonplaats?

Er kan vanuit twee invalshoeken naar deze vragen gekeken worden:

1. De informatiebehoefte vanuit de ontvanger van het bericht
2. De opslag van informatie in de database.

Vanuit de informatiebehoefte van een naïeve gebruiker is het voor de hand liggend om in een adres ook de historie van de vernamingen en vernummeringen te laten zien. Bij het samenstellen van berichten vanuit een genormaliseerde database leidt dit wel tot complexe queries om berichten samen te stellen. Er moet namelijk worden nagegaan of er relevante historie is vastgelegd bij de gerelateerde van een platgeslagen relatie. De berichtdefinities in bg0310 hebben het belang van de informatiebehoefte van een naïeve gebruiker zwaarder laten wegen dan de complexiteit bij het samenstellen. Er is bijvoorbeeld binnen AOA en NPS ook historie gedefinieerd voor de element openbareRuimtenaam en woonplaatsnaam. Mijn voorstel is dan ook om de huidige definities te handhaven. Het is aan de bouwers van vraag/antwoord systemen om in deze functionaliteit te voorzien door hetzij hun database te denormaliseren, hetzij de complexere queries uit te voeren.

Op dit moment zijn er systemen die een adres in een tabel implementeren zonder relaties naar een OpenbareRuimte en Woonplaats tabel en zelfs systemen die adresgegevens vastleggen als attributen bij een persoon. Voor deze systemen is het plezierig wanneer een vernaming van een adres wordt doorgegeven als een kennisgeving voor adres of persoon en niet als een wijziging in de openbareRuimte of Woonplaats. Het nadeel van het zo doorgeven van een vernaming van een woonplaats is natuurlijk wel dat niet volstaan kan worden met één bericht voor Woonplaats, maar dat er berichten gestuurd moeten worden voor alle objecten waarin die woonplaatsnaam voorkomt. In de gemeente Amsterdam leidt een wijziging van 'Amsterdam' naar 'Amsterdamned' tot ruwweg twee miljoen berichten (ook persoonsberichten, omdat de woonplaatsnaam ook daarin is platgeslagen) die ook nog eens naar tientallen afnemers moeten. Leggen we de last voor de oplossing van deze problematiek bij de ontvangende systemen of bij de verzendende systemen en de brokers?

Het sectormodel bg0310 specificeert nergens expliciet dat systemen verplicht zijn kennisgevingen voor Woonplaats en OpenbareRuimte te ondersteunen. Het lijkt goed om hetzij deze verplichting te expliciteren, hetzij te expliciteren voor welke platgeslagen gegevens wijzigingen in niet-strikt identificerende gegevens ook moeten worden doorgegeven met kennisgevingen voor de entiteit waarin de relatie is platgeslagen. Mijn voorstel is om gezien de problematiek met de vele berichten ervoor te kiezen dat een weinig voorkomende wijziging in de woonplaatsnaam niet wordt doorgegeven in de entiteiten waarin de woonplaatsnaam is platgeslagen, maar uitsluitend via een bericht voor woonplaats. Voor de vernaming van een openbareRuimte of de vernummering van een adres slaat de balans eerder door naar het gebruiksgemak en daar wordt expliciet voorgeschreven dat een wijziging moet worden doorgegeven in alle objecten waarin de openbareRuimtenaam, het huisnummer e.d. zijn opgenomen.

In hoofdstuk 3 zal bij het platslaan van relaties expliciet worden aangegeven hoe met deze problematiek moet worden omgegaan in het sectormodel zkn0310.

2.2 Het definiëren van relaties en hun gerelateerden

Bij het definiëren van berichtentiteitstypen wordt de netwerkstructuur van het semantisch model omgezet naar een hiërarchische structuur. De best practice voor kennisgevingen is dat in een gerelateerde alleen de kerngegevens worden opgenomen. Voor vraag/antwoord berichten is leidend welke informatie een gebruiker in één keer wil kunnen opvragen. De nu veelal gehanteerde best practice is om in een vraag/antwoord bericht als gerelateerde het type uit de kennisgeving te gebruiken. Hierdoor zijn alle attribuutsoorten en relatiesoorten van de gerelateerde beschikbaar in vraag/antwoord berichten en van gerelateerden van gerelateerden alleen de identificerende gegevens.

Voor kennisgevingberichten spelen meer afwegingen een rol:

1. Wat mag de kennisgeving met de gerelateerde doen?

Een gerelateerd object mag volgens de StUF-standaard zonder meer vanuit een relatie worden toegevoegd. Waar de bovenstaande best practice wordt gevolgd, zullen alleen de kerngegevens worden opgenomen en zal het record naderhand nog moeten worden aangevuld met de overige gegevens via vraag/antwoord of vraag-om-synchronisatie bericht. De zender van het bericht kan via het attribute verwerkingssoort aangeven of hij verwacht dat de gerelateerde al bij de ontvanger bekend is (verwerkingssoort 'I') of niet (verwerkingssoort 'T'). Het lijkt zinnig om alleen verwerkingssoort 'I' toe te staan voor de gerelateerde, indien het gewenst is dat de gerelateerde aan het ontvangende systeem wordt aangeboden, voordat de relatie gelegd wordt. Dit is het geval, als het aannemelijk is dat de gerelateerde in hetzelfde systeem wordt onderhouden als de topfundamenteel vanwaaruit de relatie ligt. Wanneer de gerelateerde in een ander systeem kan worden onderhouden, dan lijkt het niet noodzakelijk om de gerelateerde eerst aan te bieden en is het voldoende om alleen de identificerende gegevens van de gerelateerde vast te leggen. Via vraag/antwoordberichten kunnen dan zonodig de gegevens van zo'n gerelateerd object worden opgehaald. Het wijzigen van een gerelateerd object is volgens de StUF-standaard alleen toegestaan, als dit expliciet in het sectormodel wordt gespecificeerd.

2. Wat is de gewenste transactiegrootte en berichtgrootte?

Technisch is er de afweging tussen veel kleine transacties (berichten) en weinig grote transacties (berichten). Tot op heden is er vanuit een streven naar robuustheid eerder gekozen voor veel kleine transacties in plaats van enkele grote transacties. De vraag is of robuustheid gegeven de huidige kracht van relationele database managementsystemen wel zo'n belangrijk criterium is. Wanneer berichten ook binaire data als bijlagen kunnen bevatten, dan wordt de vraag naar de berichtgrootte waarschijnlijk wel weer relevant. In dergelijke gevallen lijkt het verstandig te streven naar maximaal één of enkele bijlagen per bericht. Dit streven impliceert dat bijlagen eerst apart als Enkelvoudig Document (EDC) bericht verzonden worden en hetzij in datzelfde bericht aan de Zaak worden gekoppeld hetzij in een ander bericht samen met andere documenten aan de Zaak gekoppeld wordt.

3. Vanuit welke berichtentiteiten mogen relaties gewijzigd worden?

De laatste opmerking hierboven is een voorbeeld van het probleem vanuit welke berichtentiteiten relaties gewijzigd mogen worden. Er zijn drie mogelijkheden:

1. Een relatie mag gewijzigd worden vanuit beide kanten
2. Een relatie mag alleen onderhouden worden vanuit de ene kant (in ons voorbeeld Zaak)
3. Een relatie mag alleen onderhouden worden vanuit de andere kant (in ons voorbeeld Enkelvoudig Document)

Wanneer de voorkomens van de topfundamenteel en de gerelateerde onafhankelijk van elkaar kunnen bestaan, zoals bij Document en Zaak het geval is, dan lijkt het zinnig om je af te vragen vanuit welk objecttype de relatie het beste beheerd kan worden. Voor ons voorbeeld van Document en Zaak is dat mijns inziens de Zaak, omdat een zaak documenten bundelt met als invalshoek het proces waarin de documenten een rol spelen. Het lijkt dan ook verstandig om het leggen van de koppeling tussen een document en een zaak alleen toe te staan in ZAK-kennisgevingen. Omdat een Document onafhankelijk bestaat van een Zaak lijkt het goed om van het gerelateerde document alleen de kerngegevens op te nemen. Het te relateren document behoort in een ander bericht geleverd te zijn of nog geleverd te worden.

Nota Bene: Dit is een andere keuze dan ~~meerder~~ gemaakt is, waarbij de relatie wordt gelegd vanuit EDC-kennisgevingen, omdat gestreefd werd naar niet al te grote berichten. Bovenstaande werkwijze heeft als nadeel dat er altijd twee berichten nodig zijn om een relatie te leggen, namelijk een bericht met het te relateren object gevolgd door een bericht waarin de relatie gelegd wordt. Wanneer de relatie gelegd wordt vanuit het te relateren object dan is maar één bericht nodig.

Wanneer de gerelateerde voor zijn bestaan afhankelijk is van de topfundamenteel of andersom (bijvoorbeeld order/orderregel of binnen het RGBZ zaak/status), dan is de normale werkwijze vanuit de StUF-standaard om in elk geval kennisgevingen voor het afhankelijke objecttype te definiëren, omdat een gerelateerde alleen gewijzigd mag worden als dit expliciet wordt toegestaan in het sectormodel. Wanneer er daarnaast functioneel behoefte is om in één transactie een zaak met al zijn gerelateerden te kunnen overdragen, dan is het ook gewenst om de ZAKSTA-relatie te kunnen toevoegen vanuit een ZAK-kennisgeving. Let wel, als voor de afhankelijke gerelateerde historie relevant is, dan biedt de StUF-standaard niet de mogelijkheid om deze historie ook in één bericht over te dragen. StUF schrijft voor dat een gerelateerde altijd de actuele gegevens bevat. Het in één keer overdragen inclusief historie van gerelateerden is derhalve niet mogelijk. Waar historie relevant is, lijkt het daarom het beste om de relatie te leggen vanuit het afhankelijke objecttype en met van de gerelateerde alleen de kerngegevens. Daarnaast kan in het sectormodel worden voorgeschreven dat een dergelijk bericht alleen verwerkt mag worden, als de gerelateerde al bekend is bij de ontvanger. Een alternatief voor deze werkwijze is om een afhankelijk objecttype te definiëren als relatie naar een al dan niet reële gerelateerde. Voor Status binen Zaak kan Statustype als gerelateerde fungeren. Voor de relatie tussen Zaak en Status is voor dit alternatief gekozen. In een volgende versie van de StUF-standaard zou deze constructie semantisch geëxpliciteerd kunnen worden.

De best practice is om één uiteinde van de relatie aan te wijzen als eigenaar en in kennisgevingen de relatie uitsluitend te definiëren vanuit de eigenaar. Het aanwijzen van één kant van de relatie als eigenaar is van belang voor systemen die op basis van afnemerindicaties wijzigingen doorgeven. Op deze manier is altijd duidelijk vanuit welk entiteittype een relatie wordt onderhouden en wordt voorkomen dat een wijziging in een relatie dubbel wordt aangeboden. ~~W~~Alleen wanneer volstrekt niet duidelijk is welke van de twee gerelateerde objecttypen 'eigenaar' van de relatie is, ~~dan~~ lijkt ~~het goed om ervoor te kiezen dat de keuze gemaakt te mogen worden~~ de relatie vanuit beide objecttypen te onderhouden ~~kan worden~~. Van de gerelateerde worden dan alleen de kerngegevens opgenomen.

In hoofdstuk 3 zullen afwijkingen van de hierboven genoemde best practices expliciet worden vermeld met de motivatie. Als er niets staat, dan is de best practice gevolgd.

2.3 *Het expliciteren van semantiek*

De modellering van het domein in het semantisch model leidt soms tot abstracties. Een goed voorbeeld hiervan in het RGBZ is het objecttype Rol dat een relatie legt tussen een Zaak en een Betrokkene. Binnen Rol wordt in de attribootsoorten Rolomschrijving en Rolomschrijving generiek aangegeven welke rol een betrokkene speelt. Rolomschrijving generiek bevat de belangrijkste rollen.

De door StUF gedefinieerde functionaliteit voor vraagberichten kan niet goed overweg met dit soort abstracties. Bij het specificeren van de gevraagde gegevens kan je bijvoorbeeld niet aangeven dat je alleen geïnteresseerd bent in de gegevens van betrokkenen in de rol van aanvrager of uitvoerende. Je krijgt of alle betrokkenen of geen enkele betrokkene. Bij het specificeren van een selectie kan je slechts selecteren op één soort betrokkene. Je kunt bijvoorbeeld niet vragen naar de zaken met een bepaalde aanvrager en een bepaalde uitvoerende. Dit soort problemen kan opgelost worden door in de berichtentiteiten de semantiek te expliciteren door de belangrijkste rollen als verschillende relaties op te nemen. In zkn0310 is deze keuze gemaakt door voor de rollen gespecificeerd in rolomschrijving generiek aparte relaties op te nemen.

3 De StUF-entiteitstypen en hun relatiegrafieken

3.1 Besluit en besluittype



Het objecttype BESLUIT wordt geïmplementeerd als het StUF-entiteitstype BSL en het objecttype BESLUITTYPE wordt geïmplementeerd als het StUF-entiteitstype BST. De relatiesoort BESLUIT.is van.BESLUITTYPE wordt platgeslagen door binnen vraag- en antwoordberichten van BSL alle attributen van BESLUITTYPE op te nemen met uitzondering van ingangsdatum geldigheid en einddatum geldigheid. Binnen de BSL-kennisgeving wordt alleen het attribuut besluittype-omschrijving opgenomen. Alleen voor het element omschrijving van BESLUITTYPE is binnen BESLUIT formele historie gedefinieerd. Voor de overige elementen van BESLUITTYPE is binnen BESLUIT geen historie gedefinieerd. Deze historie dient te worden opgevraagd door middel van berichten voor BESLUITTYPE. De relatiesoort

BESLUITTYPE.heeft.BESLUITen wordt niet geïmplementeerd.

De relatiesoort BESLUIT.kan vastgelegd zijn als.DOCUMENT wordt uitsluitend geïmplementeerd voor de specialisatie ENKELVOUDIG DOCUMENT. Besluiten die betrekking hebben op een samengesteld document, worden gekoppeld via de enkelvoudige documenten die daarvan deel uitmaken. Voor een motivatie hiervan zie de volgende paragraaf. BESLUIT wordt gezien als de eigenaar van de relatiesoort BESLUIT.kan vastgelegd zijn als.DOCUMENT. Omdat een enkelvoudig document vaak in een ander systeem zal worden onderhouden is het niet verplicht om het document aan te bieden voordat de relatie gelegd wordt. Zowel vVerwerkingssoort 'T' is als 'T' zijn dus toegestaan in de gerelateerde.

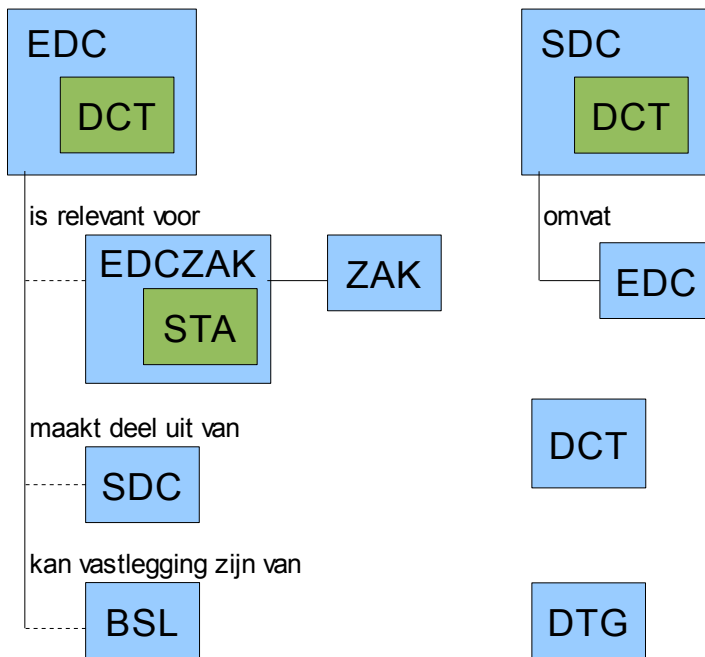
Omdat de ZAAK wordt gezien als de eigenaar van de relatie BESLUIT.is uitkomst van.ZAAK wordt deze relatie alleen gelegd in de kennisgeving voor ZAAK. Een besluit dient eerst onafhankelijk van de zaak te worden vastgelegd en daarna aan de zaak te worden gekoppeld via een ZAK-kennisgeving. De relatie kan wel worden opgevraagd in BSL-vraag/antwoordberichten.

Binnen BSL en BST worden de attribuutsoorten opgenomen zoals gedefinieerd in het RGBZ.

3.2 Document (enkelvoudig en samengesteld) en zaakdocument

Het objecttype ENKELVOUDIG DOCUMENT wordt geïmplementeerd als het StUF-entiteitstype EDC en SAMENGESTELD DOCUMENT als het StUF-entiteitstype SDC. Alleen aan een ENKELVOUDIG DOCUMENT wordt een ZAAK gekoppeld. Een SAMENGESTELD DOCUMENT kan alleen indirect aan een ZAAK gekoppeld worden. Daarom kan bij een ENKELVOUDIG DOCUMENT wel worden aangegeven dat het deel uitmaakt van een SAMENGESTELD DOCUMENT. Er is voor gekozen om alleen ENKELVOUDIGE DOCUMENTen aan een ZAAK te koppelen, omdat je bij het zoeken van een zaak op een ENKELVOUDIG DOCUMENT de zaak ook zou willen vinden als het enkelvoudige document onderdeel is van een samengesteld document. De nu gekozen implementatie dwingt dit min of meer af, omdat elk enkelvoudig document aan een zaak gekoppeld moet worden ongeacht of het deel uitmaakt van een samengesteld document of niet. Een samengesteld document wordt geacht aan een

zaak te zijn gekoppeld, als alle enkelvoudige documenten die deel uitmaken van het samengestelde document aan de zaak zijn gekoppeld. Het koppelen van een enkelvoudig document dat deel uitmaakt van een samengesteld document impliceert dus niet dat dit samengestelde document ook aan de zaak gekoppeld is. Omdat het werken met samengestelde documenten voorlopig nog geen gemeengoed is, wordt het niet als een ernstig gemis gezien dat je een samengesteld document met al zijn enkelvoudige documenten niet in één handeling kunt koppelen aan een zaak.



Er is ook voor gekozen om de generalisatie DOCUMENT niet te implementeren als een StUF-entiteitstype. Zolang alleen ENKELVOUDIGE DOCUMENTen aan een ZAAK gekoppeld kunnen worden is er geen behoefte aan dit supertype. Wanneer uit de praktijk blijkt dat het direct koppelen van een SAMENGESTELD DOCUMENT aan een ZAAK belangrijk is, dan zal dit opgenomen dienen te worden in een volgende versie van het sectormodel ZKN.

Het objecttype DOCUMENTTYPE wordt geïmplementeerd als het StUF-entiteitstype DCT. Het domein voor de attribuutsoort documenttype-omschrijving generiek wordt geïmplementeerd als het StUF-tabelentiteitstype DTG met de elementen

omschrijving, ingangsdatumObject en einddatumObject. De relatiesoort DOCUMENT.is van. DOCUMENTTYPE wordt platgeslagen door binnen vraag- en antwoordberichten van EDC en SDC de attribuutsoorten documenttype-omschrijving, documenttype-omschrijving generiek en [document](#)categorie op te nemen. De historie van documenttype-omschrijving generiek [en](#) documenttype-categorie kan niet worden opgevraagd via vraag/antwoordberichten voor EDC of SDC, maar alleen via DCT-berichten. Binnen de EDC- en SDC-kennisgeving wordt alleen de attribuutsoort omschrijving opgenomen. De relatiesoort DOCUMENTTYPE.heeft.DOCUMENTen wordt niet geïmplementeerd.

Het relatie-entiteitstype EDCZAK correspondeert met het objecttype ZAAKDOCUMENT in het RGBZ. EDCZAK wordt uitsluitend in vraag/antwoordberichten voor ENKELVOUDIG DOCUMENT geïmplementeerd, omdat ZAAK wordt gezien als de eigenaar van deze relatie. De relatie tussen een enkelvoudig document en een zaak kan dus alleen gelegd worden met een kennisgevingbericht voor ZAK. De relatiesoort ZAAKDOCUMENT.is relevant voor.STATUS wordt in EDCZAK platgeslagen door de attribuutsoorten volgnummer en omschrijving uit het bij de STATUS behorende STATUSTYPE en datumStatusGezet uit STATUS op te nemen. Voor deze platgeslagen gegevens is geen historie gedefinieerd binnen EDCZAK.

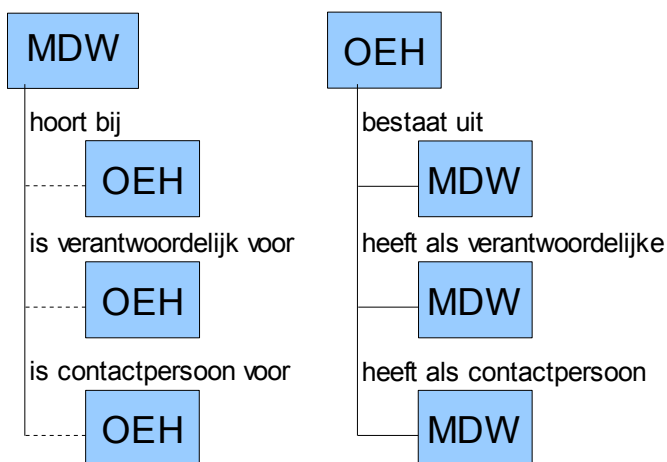
De relatiesoort DOCUMENT.kan vastlegging zijn van.BESLUIT wordt uitsluitend geïmplementeerd in vraag/antwoord berichten. Deze relatie kan uitsluitend gelegd worden via een kennisgevingbericht voor BESLUIT.

De relatiesoort ENKELVOUDIG DOCUMENT maakt deel uit van SAMENGESTELD DOCUMENT wordt in het relatie-entiteittype EDCSDC geïmplementeerd en wel uitsluitend in vraag/antwoordberichten, omdat SAMENGESTELD DOCUMENT wordt gezien als de eigenaar van deze relatie. Van het gerelateerde samengestelde document worden geen enkelvoudige documenten getoond die er deel van uitmaken. Deze dienen opgevraagd te worden in een apart vraag/antwoordbericht voor SDC. De relatie ENKELVOUDIG DOCUMENT maakt deel uit van SAMENGESTELD DOCUMENT kan alleen gelegd worden door middel van een SDC-kennisgeving via het relatie-entiteittype SDCEDC. Eerst dient in apart EDC-kennisgevingbericht het enkelvoudig document te zijn aangeboden en vervolgens kan dit al dan niet samen met andere enkelvoudige documenten in een SDC-kennisgeving aan een samengesteld document worden gekoppeld. De SDC-kennisgeving bevat van de gerelateerde uitsluitend de kerngegevens en als verwerkingssoort voor de gerelateerde is alleen 'I' toegestaan.

Het attribuutsoort documentInhoud is opgenomen in de StUF-entiteit EDC als het element met als elementnaam 'inhoud' met als complexType StUF:BinairInhoud gedefinieerd in het schema stuf0301mtom.xsd. Het sectormodel zkn0310 maakt daardoor voor het verzenden van binaire bijlagen gebruik van de protocolbinding 0301.

Binnen EDC, SDC, EDCZAK en DCT worden voor de rest de attribuutsoorten en relatiesoorten opgenomen zoals gedefinieerd in het RGBZ.

3.3 Medewerker en organisatorische eenheid



Het objecttype MEDEWERKER wordt geïmplementeerd als het StUF-entiteittype MDW en het objecttype ORGANISATORISCHE EENHEID wordt geïmplementeerd als het StUF-entiteittype OEH.

De relatiesoorten MEDEWERKER.hoort bij.ORGANISATORISCHE EENHEID, MEDEWERKER.is verantwoordelijk voor.ORGANISATORISCHE EENHEID en MEDEWERKER.is contactpersoon voor.ORGANISATORISCHE EENHEID

worden vanuit MEDEWERKER uitsluitend geïmplementeerd als vraag/antwoord bericht, omdat ORGANISATORISCHE EENHEID wordt gezien als de eigenaar van deze relaties. De relatie dient gelegd te worden door middel van kennisgevingberichten voor OEH, nadat eerst de medewerker in een aparte kennisgeving is aangeboden. De gerelateerde MDW in een OEH-kennisgeving bevat alleen de kerngegevens en alleen verwerkingssoort 'I' is toegestaan.

De relatiesoort ROL tussen ZAAK en MEDEWERKER of ORGANISATORISCHE EENHEID wordt uitsluitend geïmplementeerd vanuit ZAAK. In een kennisgevingbericht voor een ZAAK kan een medewerker of organisatorische eenheid in een bepaalde rol aan die zaak gekoppeld worden. In de vraagberichten voor ZAAK zijn ook zoekingen gedefinieerd om alle zaken te kunnen vinden, waar een medewerker in de een of andere rol bij betrokken is. Ook de relatiesoorten MEDEWERKER.is verantwoordelijke voor.ZAAKTYPE en ORGANISATORISCHE EENHEID.is verantwoordelijke voor.ZAAKTYPE worden uitsluitend geïmplementeerd vanuit ZAAKTYPE.

Ook bij ZAAKTYPE zijn zoekingen gedefinieerd om alle ZAAKTYPEEn waarvoor een bepaalde medewerker of organisatorische eenheid verantwoordelijk is te vinden.

De in het objectmodel getekende relatie tussen ORGANISATORISCHE EENHEID en VESTIGING is niet beschreven in het RGBZ en daarom tot op heden niet meegenomen in de verStUffing. Het is wenselijk dat eerst het RGBZ wordt aangepast rekening houdend met de post op het forum over de modellering van BETROKKENE, voordat verdere besluiten over de verStUffing worden genomen. Een belangrijke vraag hierbij is ook nog wie eigenaar is van deze relatie wordt of te wel vanuit welke kant de relatie wordt onderhouden. Onderhoud vanuit VESTIGING – wat het meest logisch lijkt – is alleen mogelijk, als binnen zkn0310 een eigen VES-entiteittype wordt gedefinieerd met een isEen relatie naar VES uit bg0310.

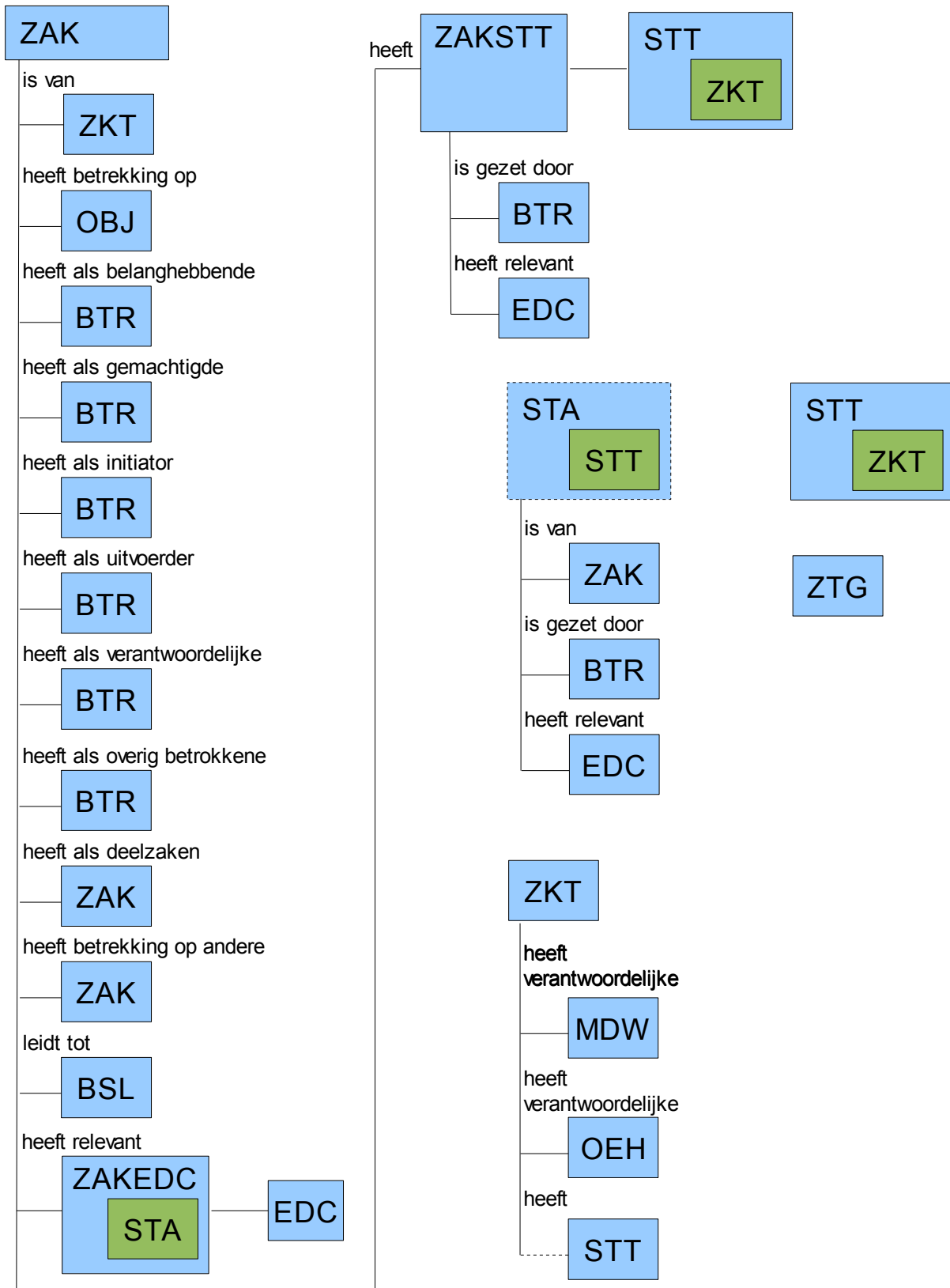
De attribuutsoorten worden geïmplementeerd zoals beschreven in het RGBZ.

3.4 Zaak, zaaktype, status en statustype

Het objecttype ZAAK wordt geïmplementeerd als het StUF-entiteittype ZAK, het objecttype ZAAKTYPE wordt geïmplementeerd als het StUF-entiteittype ZKT en het objecttype ZAAKTYPE GENERIEK wordt geïmplementeerd als het StUF-tabelentiteittype ZTG. Voor de ZAAK als gerelateerde worden de attribuutsoorten opgenomen met betrekking tot de stand van zaken van de zaakbehandeling. Attribuutsoorten met betrekking tot het proces van afhandeling zijn niet opgenomen. In ZAAK als gerelateerde worden uitsluitend de relatiesoorten ZAAK.is van.ZAAKTYPE, ZAAK.heeft betrekking op.OBJECT, ZAAK.heeft als belanghebbende.BETROKKENE, ZAAK.heeft als initiator. BETROKKENE, ZAAK.heeftAlsDeelzaak.ZAAK en ZAAK.heeft.STATUS opgenomen.

De relatie tussen ZAAK en OBJECT is in het RGBZ gemodelleerd met de relatiesoort ZAAKOBJECT met als enige attribuutsoort Relatie-omschrijving. ZAAKOBJECT wordt geïmplementeerd in het relatie-entiteittype ZAKOBJ. Van het gerelateerde object worden in een kennisgeving uitsluitend de kerngegevens opgenomen. Het is niet noodzakelijk om het gerelateerde object eerst in een kennisgeving aan te bieden. Zowel Verwerkingssoort 'T' als 'I' voor het gerelateerde object is zijn derhalve toegestaan. De relatie naar DOCUMENT als OBJECT wordt geïmplementeerd door zowel ENKELVOUDIG DOCUMENT als SAMENGESTELD DOCUMENT op te nemen als subtype van OBJECT. De door het RGBZ onderkende subtypen FUNCTIONARIS en WOZ-BELANG van OBJECT worden niet geïmplementeerd, omdat deze typen niet als fundamentele entiteitstypen zijn gedefinieerd in bg0310, maar als relatie-entiteitstypen. Een relatie-entiteittype kan geen subtype zijn. Een en ander zou een functionele aanpassing van bg0310 vergen. Het kunnen zoeken op de geometrische kenmerken van een OBJECT is nu niet geïmplementeerd, doordat het element geometrie niet is opgenomen binnen het StUF entiteittype OBJ. Dit wacht op de harmonisatie van StUF met NEN3610.

De relatie tussen ZAAK en BETROKKENE is in het RGBZ gemodelleerd met de relatiesoort ROL met een aantal attribuutsoorten. Eén van deze attribuutsoorten is 'Rolomschrijving generiek' met als waardebereik de enumeratie Belanghebbende, Gemachtigde, Initiator, Overig, Uitvoerder en Verantwoordelijk. Bij het verStUffen is ervoor gekozen om de verschillende waarden in deze enumeratie te implementeren als de relatie-entiteiten ZAKBTRBLH (belanghebbende), ZAKBTRGMC (gemachtigde), ZAKBTRINI (initiator), ZAKBTRUTV (uitvoerder), ZAKBTRVRA (verantwoordelijke) en ZAKBTROVR (Overig). Al deze relatie-entiteitstypen bevatten de overige attribuutsoorten en relatiesoorten van ROL. Dit is gedaan, omdat dan de



functionaliteit voor het selecteren van zaken op basis van de rol van de betrokkene beter werkt in StUF. Op deze manier kan op meerdere rollen tegelijkertijd geselecteerd worden. De relatie-soort ROL bij BETROKKENE.met als binnenlands correspondentieadres.ADRESSEERBAAR OBJECT

AANDUIDING is platgeslagen door de CorrespondentieAdrGrp uit bg0310 op te nemen binnen de verschillende ZAKBTR relatie-entiteitstypen met als elementnaam afwijkendCorrespondentieAdres. Deze groep bevat ook de postbusnummer of antwoordnummer gegevens. De relaties naar WOONPLAATS en LAND binnen de adresgegevens in ROL zijn platgeslagen door het opnemen van de woonplaatsnaam c.q. de landcode en de landnaam. Voor al deze platgeslagen adresgegevens is geen historie gedefinieerd. De relatie-entiteitstypen ZAKBTRINI en ZAKBTRVRA hebben kardinaliteit 0-1. De overige relatie entiteitstypen hebben kardinaliteit 0-N. Van de gerelateerde betrokkenene worden in een kennisgeving uitsluitend de kerngegevens opgenomen. Het is niet noodzakelijk om de betrokkene eerst aan te bieden, omdat de betrokkene regelmatig zal worden onderhouden in een ander systeem. Zowel verwerkingssoort 'T' is als 'T' zijn derhalve toegestaan in de gerelateerde betrokkene.

De relatiesoort ZAAK is deelzaak van. ZAAK is in de inverse richting geïmplementeerd als ZAAK. heeft als deelzaken. ZAAK in het relatie-entiteitstype ZAKZAKDEL, omdat de 'hoofd' zaak wordt beschouwd als de eigenaar van de relatie. De zaak die als deelzaak wordt gekoppeld dient eerst in een kennisgeving te worden aangeboden, omdat vanuit de 'hoofd'zaak statusinformatie over deelzaken gevolgd moet kunnen worden. Alleen verwerkingssoort 'T' is derhalve toegestaan in de gerelateerde. ~~NB: Wat mij betreft is het een discussiepunt of dit echt wenselijk is. Ik heb~~ Er is -hier ook gekozen voor het eerst leveren van de zaak, omdat het lastig te achterhalen lijkt waar informatie over een deelzaak te vinden is. Het RGBZ biedt geen voorzieningen om eenvoudig aan te geven waar een zaak technisch te vinden is.

De relatie tussen ZAAK en DOCUMENT is in het RGBZ gemodelleerd met de objectsoort ZAAKDOCUMENT met een aantal attribootsoorten en relatiesoorten. ZAAKDOCUMENT wordt geïmplementeerd in het relatie-entiteitstype ZAKEDC. De relatie ZAAK. heeft relevant. SAMENGESTELD DOCUMENT wordt niet geïmplementeerd. Deze relatie wordt indirect gelegd via de enkelvoudige documenten die deel uitmaken van het samengestelde document.

Een enkelvoudig document kan alleen aan een zaak worden gekoppeld door middel van een ZAK-kennisgevingbericht, omdat ZAAK wordt gezien als de eigenaar van de relatie. Het te koppelen enkelvoudige document hoeft niet eerst te worden aangeboden, omdat het vaak in een ander systeem zal worden onderhouden. Van de gerelateerde EDC in ZAKEDC worden uitsluitend de kerngegevens opgenomen en is zowel verwerkingssoort 'T' als 'I' zijn toegestaan. De relatiesoort ZAAKDOCUMENT. is relevant voor. STATUS wordt in ZAKEDC platgeslagen door de attribootsoorten volgnummer en omschrijving uit het bij de STATUS behorende STATUSTYPE en datumStatusGezet uit STATUS op te nemen. Voor de platgeslagen elementen is geen historie gedefinieerd.

De relatiesoort ZAAK. leidt tot. BESLUIT wordt in kennisgevingen en vraag/antwoord berichten geïmplementeerd. Een besluit dient eerst via een kennisgeving te worden aangeboden en daarna via een ZAK-kennisgeving aan een zaak te worden gekoppeld. Als verwerkingssoort voor het gerelateerde besluit is uitsluitend 'T' toegestaan.

Het objecttype STATUS wordt zowel geïmplementeerd als het relatie-entiteitstype ZAKSTT van ZAAK naar STATUSTYPE als als een afzonderlijk fundamenteel entiteitstype STA. Wijzigingen in een status kunnen uitsluitend worden doorgegeven door middel van kennisgevingen voor ZAK. De verwerkingssoort voor het gerelateerde statustype mag uitsluitend 'T' zijn. Er is voor implementatie van STATUS als het relatie-entiteitstype ZAKSTT gekozen, omdat STATUS voor zijn bestaan

afhankelijk is van ZAAK. Dankzij deze constructie kan in één synchronisatiebericht een zaak inclusief alle historie van zijn statussen worden overgedragen. In het relatie-entiteitstype ZAKSTT is formele historie met een tijdvakRelatie gedefinieerd, omdat eenzelfde status meerdere keren bij een zaak kan voorkomen. Het element beginRelatie dient dezelfde waarde te hebben als datumStatusGezet binnen STATUS. Het element eindRelatie krijgt alleen een waarde als een status ten onrechte bij een zaak is gezet. In dat geval krijgt eindRelatie dezelfde waarde als beginRelatie. In alle andere gevallen behoudt eindRelatie de waarde StUF:noValue="geenWaarde", omdat een eenmaal gezette status altijd actueel blijft.

Omdat het nogal onnatuurlijk is om statussen van verschillende zaken op te vragen in een vraag/antwoordbericht voor ZAAK, is STATUS ook geïmplementeerd als het fundamenteel entiteitstype STA met uitsluitend vraag/antwoord berichten. De relatie STATUS.is van.STATUSTYPE wordt in de vraag/antwoordberichten platgeslagen door binnen STA de attribuutsoorten statutypevolnummer en statutype-omschrijving op te nemen. Binnen STA wordt geen historie ondersteund voor het element omschrijving uit STATUSTYPE. De relatiesoort ZAAK.heeft.STATUS wordt uitsluitend geïmplementeerd in vraag/antwoord berichten voor ZAAK. Deze relatie dient gelegd te worden door middel van kennisgevingberichten voor STATUS, omdat STATUS voor zijn bestaan afhankelijk is van ZAAK. De verwerkingssoort voor de gerelateerde zaak mag uitsluitend 'I' zijn. Het moment waarop de status wordt gezet ligt vast binnen STATUS en het RGBZ specificeert dat een ZAAK meerdere STATUSsen kan hebben. Een STATUS wordt derhalve nooit beëindigd en wordt niet historisch. In de relatie ZAAK.heeft.STATUS wordt StUF:tijdvakRelatie wel opgenomen, omdat voor deze relatie formele historie relevant is. StUF:tijdvakRelatie is nodig om aan te kunnen geven dat een status onterecht gezet is. De waarde van het element beginRelatie dient gelijk te zijn aan de waarde van het element datumStatusGezet.

De relatie tussen STATUS en BETROKKENE via de relatiesoort ROL is een andere dan de relatie tussen ZAAK en BETROKKENE via de relatiesoort ROL. Uit de definitie van de relatiesoort STATUS.is gezet door betrokkene in zijn/haar.ROL blijkt dat een status alleen gezet kan worden door een betrokkene die in een rol bij de zaak voorkomt. Omdat een betrokkene in meerdere rollen bij een status kan voorkomen, is de rol waarin de status gezet wordt, ook een eigenschap van de relatie tussen STATUS en BETROKKENE. De relatie tussen STATUS en BETROKKENE wordt als het relatie-entiteitstype ZAKSTTBTR vanuit ZAKSTT geïmplementeerd in ZAK-kennisgevingen en als het relatie-entiteitstype STABTR in vraag/antwoordberichten voor STA. De rol waarin de betrokkene de status zet, wordt als het attribuutsoort Rolomschrijving opgenomen binnen het relatie-entiteitstype ZAKSTTBTR respectievelijk STABTR. De overige attribuutsoorten en relatiesoorten van ROL zijn niet relevant, omdat deze afgeleid kunnen worden van de relatiesoort ROL tussen ZAAK en BETROKKENE. Binnen ZAKSTTBTR en STABTR is alleen formele historie gedefinieerd, omdat een status precies één keer door een betrokkene in een bepaalde rol gezet kan worden. Materiële historie is derhalve niet relevant.

De relatie STATUS.heeft.relevant.ZAAKDOCUMENT wordt geïmplementeerd als het relatie-entiteitstype ZAKSTTEDC vanuit ZAKSTT in kennisgevingen voor ZAK en als STAEDC in vraag/antwoord berichten voor STA. Als gerelateerde in ZAKSTTEDC wordt EDC opgenomen met alleen de kerngegevens. Dit relatie-entiteitstype heeft verder geen attribuutsoorten, omdat het alleen gaat om een verwijzing naar de documenten relevant voor de status. De gegevens relevant voor de relatie tussen het document en de zaak zijn te vinden in het relatie-entiteitstype ZAKEDC (implementatie van de relatiesoort ZAAKDOCUMENT). Omdat het document al aan de zaak gerelateerd moet zijn, is alleen verwerkingssoort 'I' toegestaan in de gerelateerde.

Voor het overige bevat ZAK de attribuutsoorten gedefinieerd in het RGBZ voor ZAAK en bevatten STA en ZAKSTT de attribuutsoorten gedefinieerd in het RGBZ voor STATUS.

~~Voor het overige bevat ZAK de attribuutsoorten gedefinieerd in het RGBZ.~~

~~De relatiesoort ZAAK leidt tot BESLUIT wordt in kennisgevingen en vraag/antwoord berichten geïmplementeerd. Een besluit dient eerst via een kennisgeving te worden aangeboden en daarna via een ZAK kennisgeving aan een zaak te worden gekoppeld. Als verwerkingssoort voor het gerelateerde besluit is uitsluitend 'I' toegestaan.~~

Het objecttype ZAAKTYPE wordt met uitzondering van de relatiesoort ZAAKTYPE.betreft.ZAAK en geïmplementeerd als het StUF-entiteittype ZKT. Het domein voor de attribuutsoort zaaktype-omschrijving generiek wordt geïmplementeerd als het StUF-tabelentiteittype ZTG met de elementen omschrijving, ingangsdatumObject en einddatumObject. De relatie naar de verantwoordelijke MEDEWERKER en ORGANISATORISCHE EENHEID wordt opgenomen in de ZKT-kennisgevingen met in de gerelateerde alleen de kerngegevens. Het is niet noodzakelijk om de medewerker of organisatorische eenheid eerst aan te bieden. Als verwerkingssoort voor de gerelateerde zijn zowel 'I' als 'T' toegestaan. De relatiesoort ZAAKTYPE.heeft.STATUSTYPE en wordt uitsluitend geïmplementeerd in vraag/antwoord berichten, omdat een statutype voor zijn bestaan afhankelijk is van het zaaktype.

Het objecttype STATUSTYPE wordt geïmplementeerd als het StUF-entiteittype STT. De relatiesoort STATUSTYPE is van ZAAKTYPE wordt platgeslagen door het opnemen van attribuutsoort omschrijving uit ZAAKTYPE binnen STATUSTYPE. De relatiesoort STATUSTYPE.heeft.STATUS en worden uitsluitend geïmplementeerd vanuit STATUS. De relatiesoort STATUSTYPE is van ZAAKTYPE wordt platgeslagen door het opnemen van attribuutsoort omschrijving uit ZAAKTYPE binnen STATUSTYPE. De relatiesoort STATUSTYPE.heeft.STATUS en worden uitsluitend geïmplementeerd vanuit STATUS. voor het element omschrijving uit STATUSTYPE bijgehouden. De relatie STATUS is van STATUSTYPE wordt platgeslagen door binnen STA de attribuutsoorten statutypevolgnummer en statutype-omschrijving op te nemen. Binnen STA wordt geen historie en het objecttype STATUSTYPE wordt geïmplementeerd als het StUF-entiteittype STT. STA-entiteittype StUF-cHet objecttype STATUS wordt geïmplementeerd als het

~~Het objecttype STATUSTYPE wordt geïmplementeerd als het entiteittype STT. De relatie STATUSTYPE is van ZAAKTYPE wordt opgeslagen door het opnemen van de attribuutsoort omschrijving uit ZAAKTYPE in STATUSTYPE. De overige attribuutsoorten van STATUSTYPE worden opgenomen zoals gedefinieerd in het RGBZ. De relatie tussen STATUS en BETROKKENE via de relatiesoort ROL is een andere dan de relatie tussen ZAAK en BETROKKENE via de relatiesoort ROL. Uit de definitie van de relatiesoort STATUS is gezet door betrokkene in zijn/haar ROL blijkt dat een status alleen gezet kan worden door een betrokkene die in een rol bij de zaak voorkomt. Omdat een betrokkene in meerdere rollen bij een status kan voorkomen, is de rol waarin de status gezet wordt, ook een eigenschap van de relatie tussen STATUS en BETROKKENE. Deze eigenschap wordt als het attribuutsoort Rolomschrijving opgenomen binnen het relatie-entiteittype STABTR. De overige attribuutsoorten en relatiesoorten van ROL zijn niet relevant, omdat deze afgeleid kunnen worden van de relatiesoort ROL tussen ZAAK en BETROKKENE. Binnen STABTR is alleen formele historie gedefinieerd, omdat een status precies één keer door een betrokkene in een bepaalde rol gezet kan worden. Materiële historie is derhalve niet relevant.~~

~~De relatie STATUS heeft relevant ZAAKDOCUMENT wordt geïmplementeerd als het relatie-entiteitstype STAEDC in zowel kennisgevingen als vraag/antwoord berichten. In een kennisgeving wordt EDC opgenomen met alleen de kerngegevens. Dit relatie-entiteitstype heeft verder geen attribuutsoorten, omdat het alleen gaat om een verwijzing naar de documenten relevant voor de status. De gegevens relevant voor de relatie tussen het document en de zaak zijn te vinden in het relatie-entiteitstype ZAKEDC (implementatie van de relatiesoort ZAAKDOCUMENT). Omdat het document al aan de zaak gerelateerd moet zijn, is alleen verwerkingssoort 'I' toegestaan in de gerelateerde.~~

~~Voor het overige bevat STA de attribuutsoorten gedefinieerd in het RGBZ.~~

~~Voor de relatie tussen STATUS en ZAAK geïmplementeerd in het relatie-entiteitstype STAZAK is formele historie met een tijdvakRelatie gedefinieerd, omdat eenzelfde status meerdere keren bij een zaak kan voorkomen. Het element beginRelatie binnen STAZAK dient dezelfde waarde als datumStatusGezet binnen STA. Het element eindRelatie binnen STAZAK krijgt alleen een waarde als een status ten onrechte bij een zaak is gezet. In dat geval krijgt eindRelatie dezelfde waarde als beginRelatie. In alle andere gevallen behoudt eindRelatie de waarde StUF:noValue="geenWaarde", omdat een eenmaal gezette status altijd actueel blijft.~~

4 Lijst met mnemonics

BLH	Belanghebbende
BSL	Besluit
BST	Besluittype
BTR	Betrokkene
BVT	Bevat
CPR	Contactpersoon
DCT	Documenttype
DEL	Deel
DTG	Documenttype generiek
EDC	Enkelvoudig document
GMC	Gemachtigde
INI	Initiator
MDW	Medewerker
OBJ	Object
OEH	Organisatorische eenheid
OND	Onderdeel
OVR	Overig
SDC	Samengesteld document
STA	Status
STT	Statustype
UTV	Uitvoerend
VRA	Verantwoordelijk
ZAK	Zaak
ZKT	Zaaktype

ZTG Zaaktype generiek

Tabel 4.1 Mnemonics met hun betekenis